IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered). Please AMEND claims 10, 16, 18, 22, 24, 32, 37, 38, 39, 40, 79, and 80, CANCEL claims 41-74 without prejudice or disclaimer and ADD new claims 81-82 in accordance with the following:

1. (previously presented) A data modulating method of modulating m-bit source data into n-bit, wherein n m, and m and n are positive integers, code words where a minimum run length limit is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers, the data modulating method comprising:

dividing an input data stream by a predetermined length and multiplexing the input data stream using multiplexed information according to a predetermined multiplexing method to provide a multiplexed data stream;

weak DC-free run length limited (RLL)-modulating the multiplexed data stream without a separate DC control code conversion table comprising additional bits; and

providing a code stream having a minimum DC component among multiplexed, RLL-modulated code streams.

2. (previously presented) The data modulating method of claim 1, wherein the multiplexing method uses a scrambling method.

3. (previously presented) The data modulating method of claim 1, wherein the multiplexing method uses an interleaving method.

4. (previously presented) The data modulating method of claim 1, wherein the weak DC-free RLL modulation is performed by generating code words complying with a predetermined RLL condition and grouping the code words according to the predetermined RLL condition using a main code conversion table in which code words are arranged so that a stream of source words has a DC control capability and a DC control sub code conversion table which is made by taking unnecessary code words complying with the predetermined RLL condition out of the main code conversion table.

5. (previously presented) The data modulating method of claim 4, wherein the minimum run length limit "d" is 1 and the maximum run length limit "k" is 7.

6. (previously presented) The data modulating method of claim 4, wherein in the weak DC-free RLL modulation, code words of same data in main code groups and in DC control auxiliary code groups, have opposite INV values to control a direct current (DC), wherein INV is a parameter predicting a transition direction of a next code word depending on whether a number of bits of value "1" in a code word stream is odd or even.

7. (previously presented) The data modulating method of claim 4, wherein the main code conversion table comprises code word groups that satisfy conditions of $0 \leq EZ \leq 5$, $1 \leq LZ \leq 7$, $0 \leq LZ \leq 4$, and $0 \leq LZ \leq 2$, wherein EZ is end zeros indicating a number of successive zeros from least significant bits to most significant bits in a code word and LZ is lead zeros indicating a number of successive zeros from most significant bits to least significant bits.

8. (previously presented) The data modulating method of claim 7, wherein one of the code groups of the main code conversion table comprising source codes less than a minimum number of code words to be modulated is prepared by removing redundant code words from one of the code groups comprising code words greater than the minimum number so that code words greater than the minimum number are retained.

9. (previously presented) The data modulating method of claim 4, wherein the DC control sub conversion table comprises:

a group comprising code words satisfying $6 \leq EZ \leq 7$ and $LZ \neq 0$, redundant code words of a first main code group, and "1010xxxxxxxx" code words satisfying $6 \leq EZ \leq 7$ and $LZ = 0$, wherein EZ is end zeros indicating a number of successive zeros from most significant bits to least significant bits and LZ is lead zeros indicating a number of successive zeros from most significant bits to least significant bits;

a group comprising code words satisfying $6 \leq EZ \leq 7$ and $0 \leq LZ \leq 6$, code words satisfying $0 \leq EZ \leq 5$ and $5 \leq LZ \leq 6$, and redundant code words of a second main code group, the group from which "1010xxxxxxxx" code words satisfying $6 \leq EZ \leq 7$, $LZ = 0$ are removed; and

a group comprising code words complying with $6 \leq EZ \leq 7$ and $0 \leq LZ \leq 3$, code words satisfying $0 \leq EZ \leq 5$ and $LZ = 3$, and redundant code words of a third main code group.

10. (currently amended) The data modulating method of claim 9, wherein when a stream of code words ~~a~~ a, b1, and c, and a stream of code words ~~a~~ a, b2, and c make a pair and the code words b1 and b2 are code words having opposite INV characteristics in the DC control sub code conversion table, although the code word ~~a~~ a, b1, b2, or ~~c~~ c is converted due to a violation against a predetermined run length limit (boundary rule) between the stream of the code words ~~a~~ a, b1, and b2 and the stream of the code words b1, b2, and ~~c~~ c, code words are arranged in the main code conversion table and the DC control sub code conversion table so that a modulated stream of code words ~~a~~ a, b1, and ~~c~~ c and a modulated stream of code words ~~a~~ a, b2, and ~~c~~ c have opposite INV characteristics.

11. (previously presented) The data modulating method of claim 4, wherein the minimum run length limit "d" is 2 and the maximum run length limit "k" is 10.

12. (previously presented) The data modulating method of claim 1, further comprising inserting a sync pattern into the multiplexed data stream to which the multiplexed information is added and converting the multiplexed information into multiplexed IDs.

13. (previously presented) The data modulating method of claim 12, wherein the multiplexed IDs satisfy the minimum run length limit "d" of 2 and the maximum run length limit "k" of 7 to increase a minimum mark length to reduce interferential noise of a signal.

14. (previously presented) A data modulating method of modulating m-bit source data into n-bit, wherein $n \geq m$, and m and n are positive integers, code words where a minimum run length limit is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers, the data modulating method comprising:

multiplexing an input data stream divided by a predetermined length into a plurality of types of pseudo random data streams using multiplexed information of predetermined bits by applying a predetermined multiplexing method to each of the pseudo random data streams; and

RLL-modulating the plurality of types of pseudo random data streams to create a modulated code stream in which an optimal DC suppression is performed.

15. (previously presented) The data modulating method of claim 14, wherein the random data streams are generated by inconsecutively scrambling the input data stream using the multiplexed information.

16. (currently amended) The data modulating method of claim 15, wherein when s_t is multiplexed information used to multiplex the input data stream divided into ~~v~~ v data streams each having a u-byte length, a number ~~a~~ a of bits of the multiplexed information s_t is less than or equal to a number ~~m~~ m of bits of input source data, wherein ~~u~~ u and ~~v~~ v are positive integers.

17. (previously presented) The data modulating method of claim 16, wherein the generation of random data streams comprises:

performing an exclusive OR operation on the multiplexed information s_t and first m-bit data of a first code modulation unit of the plurality of types of random data streams to generate modulated data;

outputting data unchanged from a second code modulation unit to a q-1th code modulation unit without performing an exclusive OR operation;

performing an exclusive OR operation on the first code modulation unit and a qth code modulation unit to generate next modulated data; and

repeating an exclusive OR operation from the qth code modulation unit to a final code modulation unit of the input data stream.

18. (currently amended) The data modulating method of claim 17, wherein when an exclusive OR operation cycle is ~~q~~ q, where ~~q~~ q is a positive integer, an error propagation probability is reduced to 1/q.

19. (previously presented) The data modulating method of claim 14, further comprising:

dividing the input data stream into the predetermined length;

inserting a sync pattern into the multiplexed pseudo random data streams to which the multiplexed information is added in the generation of the random data streams and converting the multiplexed information into multiplexed IDs; and

comparing the plurality of types of RLL-modulated code streams to select the code stream comprising the minimum of DC components.

20. (previously presented) A data modulating method of modulating m-bit source data into n-bit, wherein $n \geq m$, and m and n are positive integers, code words where a minimum run length limit is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers, the data modulating method comprising:

multiplexing an input data stream divided by a predetermined length into a plurality of types of pseudo random data streams using multiplexed information of predetermined bits by applying a predetermined multiplexing method to each of the pseudo random data streams; and

weak DC-free RLL-modulating the multiplexed data streams without using a DC control code conversion table comprising additional bits and providing a code stream comprising a minimum of DC components among multiplexed, RLL-modulated code streams.

21. (previously presented) The data modulating method of claim 20, wherein the pseudo random data streams are generated by inconsecutively scrambling the input data stream using the multiplexed information.

22. (currently amended) The data modulating method of claim 21, wherein s_t is multiplexed information used to multiplex the input data stream divided into ~~v~~ v data streams each having a u-byte length, a number ~~a~~ a of bits of the multiplexed information s_t is less than or equal to a number ~~m~~ m of bits of input source data, wherein ~~u~~ u and ~~v~~ v are integers.

23. (previously presented) The data modulating method of claim 22, wherein the generation of the pseudo random data streams comprises:

performing an exclusive OR operation on the multiplexed information s_t and first m-bit data of a first code modulation unit of the plurality of types of pseudo random data streams to generate modulated data;

outputting data unchanged from a second code modulation unit to a q-1th code modulation unit without performing an exclusive OR operation;

performing an exclusive OR operation on the first code modulation unit and a qth code modulation unit to generate next modulated data; and

repeating an exclusive OR operation from the qth code modulation unit to a final code modulation unit of the input data stream.

24. (currently amended) The data modulating method of claim 23, wherein when an exclusive OR operation cycle is ~~q~~ q , where ~~q~~ q is a positive integer, an error propagation probability is reduced to 1/q.

25. (previously presented) The data modulating method of claim 20, further comprising:

dividing the input data stream into the predetermined length;

inserting a sync pattern into the multiplexed pseudo random data streams to which the multiplexed information is added in generation of the random data streams and converting the multiplexed information into multiplexed IDs; and

comparing the plurality of types of RLL-modulated code streams to select the code stream comprising the minimum of DC components.

26. (previously presented) The data modulating method of claim 20, wherein the weak DC-free RLL modulation is performed by generating code words complying with a predetermined RLL condition and grouping the code words according to the predetermined RLL condition using a main code conversion table in which code words are arranged so that a stream of source words has a DC control capability and using a DC control sub code conversion table which is made by taking unnecessary code words complying with the predetermined RLL condition out of the main code conversion table.

27. (previously presented) The data modulating method of claim 26, wherein the minimum run length limit "d" is 1 and the maximum run length limit "k" is 7.

28. (previously presented) The data modulating method of claim 26, wherein in the weak DC-free RLL modulation, code words of same data in main code groups and in DC control auxiliary code groups, have opposite INV values to control a DC, and INV is a parameter predicting a transition direction of a next code word depending on whether a number of bits of value "1" in a code word stream is odd or even.

29. (previously presented) The data modulating method of claim 26, wherein the main code conversion table comprises code word groups that satisfy conditions of $0 \le EZ \le 5$, $1 \le LZ \le 7$, $0 \le LZ \le 4$, and $0 \le LZ \le 2$, wherein EZ is end zeros indicating a number of successive zeros from least significant bits to most significant bits in a code word and LZ is lead zeros indicating a number of successive zeros from most significant bits to least significant bits.

30. (previously presented) The data modulating method of claim 29, wherein one of the code groups of the main code conversion table comprising source codes less than a minimum number of code words to be modulated is prepared by removing redundant code words from one of the code groups comprising code words greater than the minimum number so that code words greater than the minimum number are retained.

31. (previously presented) The data modulating method of claim 26, wherein the DC control sub conversion table comprises:

a group comprising code words satisfying $6 \le EZ \le 7$ and $LZ \ne 0$, redundant code words of a first main code group, and "1010xxxxxxxx" code words satisfying $6 \le EZ \le 7$ and $LZ = 0$;

a group comprising code words satisfying $6 \le EZ \le 7$ and $0 \le LZ \le 6$, code words satisfying $0 \le EZ \le 5$ and $5 \le LZ \le 6$, and redundant code words of a second main code group, the group from which "1010xxxxxxxx" code words satisfying $6 \le EZ \le 7$, $LZ = 0$ are removed, wherein EZ is end zeros indicating a number of successive zeros from most significant bits to least significant bits and LZ is lead zeros indicating a number of successive zeros from most significant bits to least significant bits; and

a group comprising code words complying with $6 \le EZ \le 7$ and $0 \le LZ \le 3$, code words satisfying $0 \le EZ \le 5$ and $LZ = 3$, and redundant code words of a third main code group.

32. (currently amended) The data modulating method of claim 31, wherein when a stream of code words ~~a~~ a, b1, and c, and a stream of code words ~~a~~ a, b2, and c make a pair and the code words b1 and b2 are code words having opposite INV characteristics in the DC control sub code conversion table, although the code word ~~a~~ a, b1, b2, or ~~c~~ c is converted due to a violation against a predetermined run length limit/boundary rule between the stream of the code words ~~a~~ a, b1, and b2 and the stream of the code words b1, b2, and ~~c~~ c, code words are arranged in the main code conversion table and the DC control sub code conversion table so that a modulated stream of code words ~~a~~ a, b1, and ~~c~~ c and a modulated stream of code words ~~a~~ a, b2, and ~~c~~ c have opposite INV characteristics.

33. (previously presented) The data modulating method of claim 26, wherein the minimum run length limit “d” is 2 and the maximum run length limit “k” is 10.

34. (previously presented) The data modulating method of claim 25, wherein the multiplexed IDs satisfy the minimum run length limit “d” of 2 and the maximum run length limit “k” of 7 to increase a minimum mark length to reduce interferential noise of a signal.

35. (previously presented) A data demodulating method comprising:

demodulating each **n** bit of input digital data into m-bit where n ≥ m, and m and n are positive integers, of a demodulation code unit to generate a non-inverted data stream having a predetermined length; and

inconsecutively descrambling the non-inverted data stream using multiplexed information to generate an inverted data stream.

36. (previously presented) The demodulating method of claim 35, wherein the descrambling of the non-inverted data stream comprises:

performing an exclusive OR operation on a first demodulation code unit and initial data that is the multiplexed information to generate first inverted data;

outputting a second demodulation code unit to a $q\text{-}1^{th}$ demodulation code unit without performing an exclusive OR operation;

performing an exclusive OR operation on the first demodulation code unit and a first q^{th} demodulation code unit of inconsecutive q^{th} demodulation code units to generate next inverted data; and

repeating an exclusive OR operation to the final one of the remaining q^{th} demodulation code units of the non-inverted data stream to provide the inverted data stream.

37. (currently amended) A method of arranging m-bit source data into n-bit, where n ≥ m, and n and m are positive integers, code words by confining a minimum run length limit “d” to 1 and a maximum run length limit “k” to 7, where "d" and "k" are positive integers, the method comprising:

when a code word ~~a~~ a is connected to a code word ~~b~~ b, the code word ~~a~~ a is a preceding code word, the code word b is selected from code words b1 and b2, a code stream in which the code word ~~a~~ a is connected to the code word b1 is X1, and a code stream in which the code word ~~a~~ a is connected to the code word b2 is X2, arranging the code words b1 and b2 to have opposite parameters INV predicting a transition of a next code depending on whether a number of bits of value "1" in a code word is odd or even; and

when the code word ~~a~~ a is connected to the code word b1 or b2, although the code word ~~a~~ a, b1, or b2 is modulated into another type of code word according to a boundary rule, arranging the code streams X1 and X2 to have opposite parameters INV.

38. (currently amended) The method of claim 37, wherein when a number of bits of value "0" between the code word **a** and the code word b1 and between the code word ~~a~~ a and the code word b2 is less than the minimum run length limit "d"=1, the code word ~~a~~ a, b1, or b2 is modified according to the boundary rule to arrange the code words ~~a~~ a, b1, and b2 so that a number of bits of value "0" between the modified code word **a** and the modified code word b1 and between the modified code word ~~a~~ a and the modified code word b2 is greater than or equal to the minimum run length limit "d"=1 and less than or equal to the maximum run length limit "k"=7.

39. (currently amended) The method of claim 37, wherein the code word ~~a~~ a in the code stream X1 and the code word ~~a~~ a in the code stream X2 are each converted into another type of code word according to the boundary rule to have the same INV value so that the code streams X1 and X2 have opposite INV values due to INV values of the code words b1 and b2.

40. (currently amended) A method of arranging m-bit source data into n-bit, wherein n ≥ m, and n and m are positive integers, code words by confining a minimum run length limit "d" to 1 and a maximum run length limit "k" to 7, where "d" and "k" are positive integers, the method comprising:

when a code word ~~b~~ b is connected to a code word ~~c~~ c, the code word ~~b~~ b is a preceding code word, the code word ~~b~~ b is selected from code words b1 and b2, a code stream in which the code word b1 is connected to the code word ~~c~~ c is Y1, and a code stream in which the code word b2 is connected to the code word ~~c~~ c is Y2, the code words b1 and b2 are arranged to have opposite parameters INV predicting a transition of a next code depending on whether a number of bits of value "1" in a code word is odd or even; and

when the code word b1 or b2 is connected to the code word ~~c~~ c, although the code word b1, b2, or ~~c~~ c is modulated into another type of code word according to a boundary rule, arranging the code streams Y1 and Y2 to have opposite parameters INV.

41. (cancelled)

42. (cancelled)

43. (cancelled)

44. (cancelled)

45. (cancelled)

46. (cancelled)

47. (cancelled)

48. (cancelled)

49. (cancelled)

50. (cancelled)

51. (cancelled)

52. (cancelled)

53. (cancelled)

54. (cancelled)

55. (cancelled)

56. (cancelled)

57. (cancelled)

58. (cancelled)

59. (cancelled)

60. (cancelled)

61. (cancelled)

62. (cancelled)

63. (cancelled)

64. (cancelled)

65. (cancelled)

66. (cancelled)

67. (cancelled)

68. (cancelled)

69. (cancelled)

70. (cancelled)

71. (cancelled)

72. (cancelled)

73. (cancelled)

74. (cancelled)

75. (previously presented) A computer-readable medium having computer-executable instructions for performing operations of a data modulating method of modulating m-bit source data into n-bit, wherein n ≥ m and m and n are positive integers, code words where a minimum run length limit is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers, the operations comprising:

dividing an input data stream by a predetermined length and multiplexing the input data stream using multiplexed information according to a predetermined multiplexing method to provide a multiplexed data stream;

weak DC-free run length limited (RLL)-modulating the multiplexed data stream without a separate DC control code conversion table comprising additional bits; and

providing a code stream having a minimum of DC components among multiplexed, RLL-modulated code streams.

76. (previously presented) A computer-readable medium having computer-executable instructions for performing operations of a data modulating method of modulating m-bit source data into n-bit, wherein n ≥ m and m and n are positive integers, code words where a minimum run length limit is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers, the operations comprising:

multiplexing an input data stream divided by a predetermined length into a plurality of types of pseudo random data streams using multiplexed information of predetermined bits by applying a predetermined multiplexing method to each of the pseudo random data streams; and

RLL-modulating the plurality of types of pseudo random data streams to create a modulated code stream in which an optimal DC suppression is performed.

77. (previously presented) A computer-readable medium having computer-executable instructions for performing operations of a data modulating method of modulating m-bit source data into n-bit, wherein n ≥ m and m and n are positive integers, code words where a minimum run length limit is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers, the operations comprising:

multiplexing an input data stream divided by a predetermined length into a plurality of types of pseudo random data streams using multiplexed information of predetermined bits by applying a predetermined multiplexing method to each of the pseudo random data streams; and

weak DC-free RLL-modulating the multiplexed data streams without using a DC control code conversion table comprising additional bits and providing a code stream comprising a minimum of DC components among multiplexed, RLL-modulated code streams.

78. (previously presented) A computer-readable medium having computer-executable instructions for performing operations of a data demodulating method, the operations comprising:

demodulating each **n** bit of input digital data into m-bit, where n ≥ m and m and n are positive integers, of a demodulation code unit to generate a non-inverted data stream having a predetermined length; and

inconsecutively descrambling the non-inverted data stream using multiplexed information to generate an inverted data stream.

79. (currently amended) A computer-readable medium having computer-executable instructions for performing operations of a method of arranging m-bit source data into n-bit, where n ≥ m and n and m are positive integers, code words by confining a minimum run length limit “d” to 1 and a maximum run length limit “k” to 7, where "d" and "k" are positive integers, the operations comprising:

when a code word ~~a~~ a is connected to a code word ~~b~~ b, the code word ~~a~~ a is a preceding code word, the code word b is selected from code words b1 and b2, a code stream in which the code word ~~a~~ a is connected to the code word b1 is X1, and a code stream in which the code word ~~a~~ a is connected to the code word b2 is X2, arranging the code words b1 and b2 to have opposite parameters INV predicting a transition of a next code depending on whether a number of bits of value “1” in a code word is odd or even; and

when the code word ~~a~~ a is connected to the code word b1 or b2, although the code word ~~a~~ a, b1, or b2 is modulated into another type of code word according to a boundary rule, arranging the code streams X1 and X2 to have opposite parameters INV.

80. (currently amended) A computer-readable medium having computer-executable instructions for performing operations of a method of arranging m-bit source data into n-bit, wherein n m and n and m are positive integers, code words by confining a minimum run length limit "d" to 1 and a maximum run length limit "k" to 7, where "d" and "k" are positive integers, the operations comprising:

when a code word ~~b~~ b is connected to a code word ~~c~~ c, the code word ~~b~~ b is a preceding code word, the code word ~~b~~ b is selected from code words b1 and b2, a code stream in which the code word b1 is connected to the code word ~~c~~ c is Y1, and a code stream in which the code word b2 is connected to the code word ~~c~~ c is Y2, the code words b1 and b2 are arranged to have opposite parameters INV predicting a transition of a next code depending on whether a number of bits of value "1" in a code word is odd or even; and

when the code word b1 or b2 is connected to the code word ~~c~~ c, although the code word b1, b2, or ~~c~~ c is modulated into another type of code word according to a boundary rule, arranging the code streams Y1 and Y2 to have opposite parameters INV.

81. (new) A data modulating method, comprising:

modulating source data into code words where a minimum run length is confined to "d" and a maximum run length limit is confined to "k", wherein "d" and "k" are positive integers,

wherein the modulating operation includes changing a code word a into another code word so that a sum of an EndZero of the code word a and a LeadZero of a code word b is greater than or equal to the minimum run length limit and is less than or equal to the maximum run length limit in a case in which the sum of the EndZero of the code word a and the LeadZero of the code word b is less than the minimum run length limit or is greater than the maximum run length limit, wherein the code word b is connected to the code word a, the code word a is a preceding code word, the EndZero is a number of successive zeros from least significant bits (LSBs) of the code word a to most significant bits (MSBs) and the LeadZero is a number of successive zeros from MSBs of the code word b to LSBs.

82. (new) The method of claim 81, wherein the minimum run length limit "d" is 1 and the maximum run length "k" is 7, the changing comprises changing the code word a into a code word in which the LSB of the code word is "0" when the EndZero of the code word a is "0" and the LeadZero of the code word b is "0".